

Hardware Prototyping of Iris Recognition System: A Neural Network Approach

Florence Choong Chiao Mei, Mamun bin Ibne Reaz, Tan Ai Leng and Faisal Mohd-Yasin

Faculty of Engineering (FOE),
Multimedia University
Jalan Multimedia,
63100 Cyberjaya,
Selangor, Malaysia.

Received Date: 24th October 2005 Accepted Date: 4th July 2006

ABSTRACT

Iris recognition, a relatively new biometric technology, possesses great advantages, such as variability, stability and security, making it to be the most promising method for high security environments. A novel hardware-based iris recognition system is proposed in this paper, which consists of two main parts: image processing and recognition. Image processing involves histogram stress, thresholding, cropping, transformation and normalizing that is performed by using Matlab. Multilayer perceptron architecture with backpropagation algorithm is employed to recognize iris pattern. The entire architecture was modeled using VHDL, a hardware description language. The approach obtained a recognition accuracy of 98.5%. The design was successfully implemented, tested and validated on Altera Mercury EP1M120F484C5 FPGA utilizing 4157 logic cells and achieved a maximum frequency of 121.87 MHz. This novel and efficient method in hardware, based on FPGA technology showed improved performance over existing approaches for iris recognition.

Keywords: Neural Network, IRIS recognition, VHDL, FPGA

ABSTRAK

Teknik pengenalanpastian iris merupakan satu teknologi biometrik yang masih baru. Teknologi ini mempunyai banyak kelebihan seperti kepelbagaian, kestabilan dan unsur keselamatan menjadikan ia kaedah yang paling berkesan untuk situasi yang memerlukan ciri keselamatan yang ketat. Sistem perkakasan yang baru dan unik untuk mengenalpasti iris diperkenalkan di dalam jurnal ini. Sistem ini terbahagi kepada dua bahagian utama: pemprosesan dan pengenalanpastian imej. Pemprosesan imej merangkumi langkah-langkah seperti memperbaiki kontra imej, mengeluarkan bahagian mata yang tidak diperlukan, mengekstrak dan mengumpul data yang diperlukan daripada imej, mengubah bentuk imej dan akhir sekali menormalisasikan

imej. Kesemua langkah-langkah ini dilaksanakan dengan menggunakan Matlab. Reka bentuk yang dipilih untuk mengenalpasti iris ialah Multilayer perceptron dengan algoritma backpropagation. Keseluruh reka bentuk ini telah dilaksanakan dengan menggunakan VHDL. Teknik ini telah merekod bacaan ketepatan sebanyak 98.5% dalam mengenalpasti iris. Reka bentuk ini telah berjaya direalisasikan, diuji dan disahkan dengan menggunakan Altera Mercury EP1M120F484C5 FPGA. Ia telah menggunakan sebanyak 4157 sel logik dan boleh beroperasi pada frekuensi maksima iaitu pada 121.87 MHz. Sistem perkakasan yang baru dan efisien ini yang berdasarkan teknologi FPGA, telah menghasilkan keputusan yang lebih baik apabila dibandingkan dengan teknik-teknik yang sedia ada.

Kata kunci: Neural Network, IRIS recognition, VHDL, FPGA

INTRODUCTION

Biometric personal identification has been largely motivated by the increasing requirement for security in a networked society. Unlike traditional token-based methods for personal identification, biometrics (Jain et al. 1999) employs various physiological or behavioral characteristics, such as fingerprints, face, facial thermograms, iris, retina, gait, palm-prints and hand geometry to accurately identify each individual. As a physiological biometric, iris recognition aims to identify persons using iris characteristics of human eyes.

Iris recognition is receiving increased attention due to its high reliability (Daugman 2001). The human iris is an annular region between the black pupil and the white sclera. The texture of iris are unique to each subject (Daugman 2001; Daugman 1993 & Daugman 2003) and has an extraordinary structure that provides many interlacing minute characteristics such as freckles, coronas, stripes, furrows, crypts and so on. Few research works (Daugman 1993 & Wildes et al. 1996) have also stated that the iris is essentially stable through a person's life. Furthermore, since the iris is an internal organ as well as externally visible, iris-based personal identification systems can be non-invasive to their users (Daugman 2001; Daugman 1993; Daugman 2003 & Wildes et al. 1996), which is greatly important for practical applications. All these desirable properties (i.e., uniqueness, stability, flexibility, reliability and non-invasiveness) make iris recognition suitable for highly reliable personal identification.

There are few works that have been done in this area. In 1987, Flom and Safir first proposed the concept of automated iris recognition, though an attempt to using the iris as an approach to human identification can be traced back to as early as 1885 (Bertillon 1885). In 1993, 2001 and 2003, Daugman used multi-scale quadrature wavelets to extract texture structure information of the iris

to generate a 2048-bit iris code and compared the difference between a pair of iris representations by computing their Hamming distance via the XOR operator. The algorithms for iris recognition developed by Daugman formed the basis for current iris recognition systems. However, the system of Daugman concentrated on ensuring that repeated image captures produced irises on the same location within the image, had the same resolution, and were glare-free under fixed illumination. Boles and Boashash (1998) have calculated zero-crossing representation of 1-D wavelet transform at various resolution levels of a virtual circle on an iris image to characterize the texture of the iris. Iris matching was based on two dissimilarity functions. It is tolerant to illumination variation, but only feature extraction and matching algorithm were considered.

The usage of the fourth-level high frequency information of an iris image's 2-D Haar wavelet transform for feature extraction was employed by Lim et al. (2001). The fourth-level details of an image's wavelet decomposition contain essentially very low frequency information. Their method did not effectively exploit middle frequency components of the iris which play an important role in recognition as well (Daugman 2001). Wildes et al. (1996) proposed a prototype system based on automated iris recognition, which registered iris image to a stored model, filtered with four resolution levels and exploited spatial correlations and Fisher linear discrimination for pattern matching. This system is very computationally demanding. In another work, global texture features of the iris were extracted by means of well-known Gabor filters at different scales and orientations (Ma et al. 2002 Vol. I). Based on the experimental results and analysis obtained, the study further constructed a bank of spatial filters, whose kernels are suitable for iris recognition, to

represent local texture features of the iris and thus achieved much better results (Ma et al. 2002 Vol. II). The drawback of this method is the size of the kernel influences the flexibility and precision of the spatial filters. The size of the kernel is limited by the processing capability of the computer system.

Most existing methods for iris recognition generally used small image sets for performance evaluation, and only the method by Daugman in 2001 and 2003 has been tested on a larger image database involving over 200 subjects. Apart from the mentioned drawbacks with previous works, all the mentioned methods are based on software implementation. Software simulations are useful for investigating the capabilities of design models and creating new algorithms; but hardware implementations remain essential for taking full advantage of parallelism and provide an increase in speed and performance. Recognition systems developed in software have several drawbacks. A limiting factor is that the size of a recognition system is limited by the size of the system that runs the code for the recognition system.

In this work, a hardware-based iris recognition system implemented on a Field Programmable Gate Array (FPGA) is proposed. FPGA is one of the most sophisticated types of programmable logic device. It provides a much higher speed as compared to software simulation (Mohd-Yasin et al. 2003). FPGA was chosen because it greatly reduces the size, offers a higher reliability, improved security, higher performance, higher accuracy and also the ability to add on features easily (Mohd-Yasin et al. 2003). In addition, the design cycle is significantly reduced, where all levels of hardware descriptions can be verified through simulations. The design can then be synthesized to gate-level netlists without further effort (Huang et al. 2002). FPGA offers the benefits of high integration levels without the risks or expenses of semi-custom and custom IC development. FPGA devices can be reconfigured to change logic function while resident in the system. This capability offers a new degree of freedom not available with the other types of logic. By using FPGA, hardware can be changed as easily as software. Design updates or modifications are easy, and can be made to products already in the field. An FPGA can be reconfigured to perform different functions at different times. These advantages have made FPGAs very popular for system prototyping.

The architecture employed to perform iris recognition was the multilayer perceptron (MLP) architecture with backpropagation algorithm. It is the most frequently used architecture and learning algorithm in the field of neural network (NN). The rationale underlying the application of NN techniques for iris recognition is that NN possesses efficient pattern recognition capability and their ability to handle noisy data. It relies on parallel data processing and is a technologically powerful and flexible tool. NNs can be used to solve complex problems with a comparatively small amount of effort (Florence et al. 2004). Apart from that, the use of FPGA to implement the architecture of the NN is especially attractive in enhancing the capabilities of NN. As NNs are parallel structures, real time processing can be achieved through hardware implementation. FPGA allows all of the system calculations to be done directly in parallel. Thus contributing to a significant increase in performance compared to a system implemented in software where the numerous internal calculations are compiled into multiple individual commands for a processor to complete sequentially.

Based on the discussions above, it is clear that the method used in this work employing hardware design results in a recognition system with a significant increase in speed, performance, reliability, security, accuracy and a reduction in size. The novel iris recognition system in hardware allows high operating frequencies with only modest processing power as well as simplified design in hardware and software. This research also presents a new algorithm using VHSIC Hardware Description Language (VHDL) that is fast, simple and easy to debug or make changes in. These added benefits coupled with a short design cycle definitely contribute to the area of iris recognition system.

DESIGN METHODOLOGY

The construction of the iris recognition system is based on the conceptual block diagram as shown in Figure 1. The objective of image processing is to improve the pictorial information for human interpretation and processing. An iris image contains not only the region of interest (iris) but also some unwanted parts. Moreover, a change in the camera-to-eye distance may result in the possible variation in the size of the same iris. In addition, the brightness is not

uniformly distributed because of non-uniform illumination. Therefore, before the process of feature extraction of the iris, the image needs to be preprocessed to localize and normalize the iris to reduce the influence of the factors mentioned above. Matlab was used to read the file containing the iris images, where it can display each pixel of the iris images.

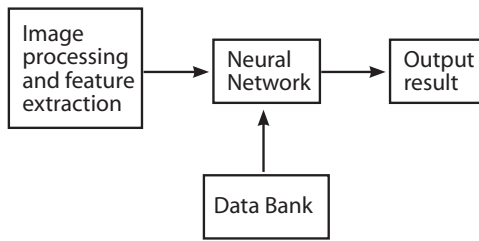


FIGURE 1. Block diagram of the iris recognition system

First, the characteristic of the image was observed by plotting the histogram of the image as shown in Figure 2. It shows that most of the pixel value ranges from 30 to 90. There are some pixels which are very dark and very bright as compared to the average level of the image. The very bright pixel is due to the reflection and the very dark pixel is the pixel in the pupil. Thus, it is safe to group these two types of pixel together. From the histogram, these pixels have a small probability in the image. By calculating the pixel from the image histogram, most of these pixels have probability less than 0.002%. We group this entire very dark and very bright pixels with probability less than 0.002% into one level and the rest of the pixels has probability more than or equal to 0.002%.

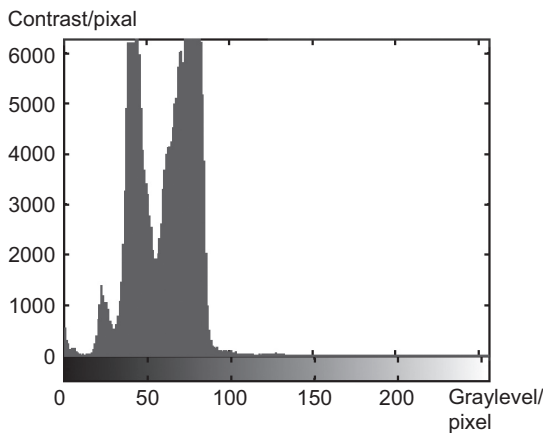


FIGURE 2. Histogram of the image of the iris

Histogram equalization is then performed if the image contrast is low. This is followed by changing gray level images to binary images. The center coordinate and radius of the iris is then located. The method employed compared with a moving circle to change the radius slowly until the radii match the iris radius of the iris. The result returned the center coordinate and the radius of the circle. Next, the iris is reconstructed from the original image. Cropping is performed on the iris to extract useful image information and to construct it into a smaller image. To crop the image, information such as the initial coordinate and size of the rectangle containing the image is supplied. Finally, the pupil of the iris is eliminated and the maximum size of the pupil is obtained. The pupil is cropped with the maximum size of the pupil for the entire image.

The NN was implemented using MLP architecture with backpropagation algorithm. MLP is a feed-forward, layered network, containing one input layer, at least one hidden layer, and one output layer. The backpropagation algorithm is employed to train the MLP that modifies network weights to minimize the mean squared error between the desired and actual outputs. The network is configured to enable any combination of number of input neurons, hidden neurons and output neurons. In this paper, a three layer NN is used. The input layer consists of three neurons; hidden layer consists of two neurons and the output layer consisting of one neuron. The rationale for choosing a smaller architecture is to reduce the complexity of the system, consume less area, faster, but at the same time able to produce good results. This can be seen in the results section. The hidden layer and output layer neurons utilized the sigmoid transfer function to generate an output. The random number generator and normalization circuit is added to initialize the weights of the NN and to process the input data to a suitable range. The network parameters in the learning algorithm formulation were determined using Equations (1) and (2).

$$\text{Error, } e_k = (o_k - d_k) \quad (1)$$

where o_k is the actual output of a neuron and d_k is the desired output.

$$\text{Squared error, } E = \left(\frac{1}{2}\right) \times \sum_{k=1}^K e_k^2, \quad (2)$$

where e_k is the error, the factor of $\frac{1}{2}$ is to simplify the computation in the later stage and K is the total number of layers. The weights between the layers at time $(t+1)$ is calculated using Equation (3),

$$W_j^k(t+1) = W_j^k(t) + \Delta W_j^k(t) \quad (3)$$

where j is the neuron number and $\Delta W_j^k(t)$ is the weight adjustment using gradient descent rule determined using Equation (4).

$$\Delta W_j^k(t) = -\eta_{jk} \frac{\partial E}{\partial W_j^k(t)} \quad (4)$$

where η_{jk} is the learning rate for weights between the layers.

Finally, the output of neuron at k -layer is given by Equation (5).

$$O_k = f^k(A^k) \quad (5)$$

where f^k is the activation function and

$$A^k = \sum_j (O_j W_j^k)$$

The NN is trained using vector features of different iris images from the data bank. The vectors are in pixel form. The image size was 300×300 pixels. It was not suitable to be fed into the network as the data is too large and will consume a lot of computation time. The size of each iris is not the same too. Therefore, the iris images were transformed into the amplitude spectrum of the images. The amplitude spectrum of the image gives the correlation of the pixel. The iris image was transformed into a rectangular form by taking the pixel in 1800 as the initial point and turned clockwise until it reaches the initial point. Then, Discrete Fast Fourier Transform is performed on the image. The original transformation of Discrete Fast Fourier Transform is to change the representation to polar coordinate. The amplitude spectrum of the image after the polar transformation is shown in the Figure 3. The spectrum shows that most of the information is assembling in the middle of the image. The middle neighborhood data of the amplitude spectrum has significant value when compared with the maximum amplitude. This neighborhood data is extracted from the amplitude spectrum.

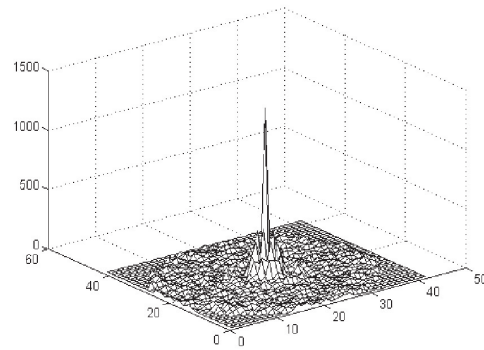


FIGURE 3. Amplitude spectrum of polar transform

The amplitude is then normalized. The image is resized by using zero order prior to feeding into the network. The enhancement is important in order to retain the entire image structure while resizing the image to a smaller size. Figure 4 shows an amplitude spectrum that was cropped and normalized. The output of the NN block is fed to the output results block. The purpose of this block is to compare the recognition results of the iris images obtained in the testing phase with the actual trained iris. If both of them are matched, then the output will display the number of the particular iris or else it will display 0.

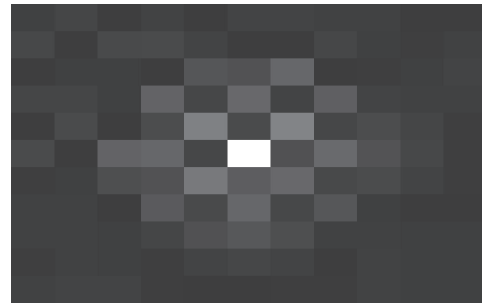


FIGURE 4. Amplitude spectrum after cropping and normalization process

SOFTWARE IMPLEMENTATION

The whole design is described using IEEE-compliant VHDL language. Hardware description language was developed to describe how hardware behaves. Optimization to the VHDL code was performed to reach a much higher speed. A top-down design approach is employed where the design is divided into five entities in which four of them are linked by top-entity, MATCHING.

The four entities are INPUT, HIDDEN LAYER, OUTPUT LAYER and BACKPROPAGATION LAYER.

Figure 5 illustrates the structural view of the system. Each distinct module performs a specific function as stated below:

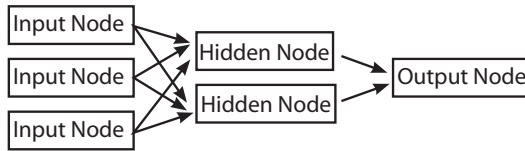


FIGURE 5. Structural View of the System

- (a) **INPUT:** The input layer consists of three neurons. The purpose of the input layer is to act as a buffer between the input data consisting of the iris images and the hidden layer neurons. The three neurons accept the vector of the iris images in pixel form.
- (b) **HIDDEN LAYER:** The hidden layer consists of two neurons that utilize a sigmoid transfer function. The developed NN is divided into three modes of operation, which are random weight generation mode, training mode and testing mode. The random weight generation mode generates random weights for each neuron. It uses a random number from 0 to 1024 to generate the desired number using Equations (6)-(8):

$$X(n) = A + (B - A) * R(n) \quad (6)$$

where,

A and B are the limits of the interval

X is a random number between A and B

R is a random number between 0 and 1

Given interval A = -1024 and B = 1024

Thus, Equation (6) is reduced to,

$$X = -1024 + (2048 * R) \quad (7)$$

In this work, the network is fed with random numbers ranging from 0 and 1024. Hence, the following formula is applied to the network instead of Equation (7).

$$\begin{aligned} X &= -1024 + (2048 * R / 1024) \\ &= -1024 + (2 * R) \end{aligned} \quad (8)$$

The value of 1024 is chosen because it is a power factor of 2. This is important since only values of power factor of 2 can be used in the division arithmetic operation for synthesis.

In the second mode of operation, the forward mode, the network propagates forward its inputs to produce outputs. Each of the hidden layer neurons multiplies each of the three input data attributes by a unique internal weight generated

by the random weights generator. These three results are added together and applied to the internal transfer squashing function that is the sigmoid function. The output of the transfer function is the activation output of the neuron. The output of the each hidden layer neurons is fed to the output layer neurons. In this mode, the network is used purely in the testing phase to perform iris recognition.

The third mode, the backward propagate mode is used to train the NN. The back propagation learning algorithm is applied in this process. This backpropagates the error computed by the network in order to adjust the weights and the threshold to minimize the error.

- (c) **BACKPROPAGATION LAYER:** The back propagation is active in the third mode. It is located after the output layer neuron to calculate the output error and adjust the weights according to the back propagation learning algorithm. The adjusted weights are fed back to the hidden layer neurons.

- (d) **OUTPUT LAYER:** The output layer consists of one neuron that utilizes a sigmoid transfer function. The output layer neurons multiply each of the two input data values from the hidden layer neurons by a unique internal weight generated by the random weights generator. The two results are added together and applied to the internal sigmoid transfer function. The output is the activation of the output layer neuron, which represents the activation of the entire NN chip. This layer performs the same modes of operation, random weights generation, normalization method and piecewise linear approximation of the sigmoid function as the hidden layer.

HARDWARE IMPLEMENTATION

Hardware implementation is the unique abstract of this work and the compilation process is repeated with different synthesis options in order to properly trade-off between area and speed. The project was successfully configured and downloaded to the Mercury EP1M120F484C5 FPGA, tested and validated. The Mercury EP1M120F484C5 device features 4800 logic cells. Altera's Quartus II version 4.0 is the software that was used in the coding, design, compilation and testing of the entire project. The project utilized a total of 4157 logic cells. The system could run at a very high frequency (F_{max}) of 121.87 MHz.

RESULTS AND DISCUSSIONS

An extensive training and testing was conducted in order to demonstrate the efficiency and feasibility of the proposed method. The overall recognition accuracy is obtained by performing timing simulation. The system produced a recognition accuracy of 98.5% using test and training sets consisting of 1000 data samples. The percentage of correct recognition is calculated using Equation (9).

% of accuracy =

$$100 - \left[\sum_{i=1}^N ((X - Y) / X) \times 100 \right] \div N$$

$$= 98.5\%$$
(9)

where,

N = No. of samples,

X = Actual iris vector,

Y = Predicted iris vector

The system functions in two different modes, namely training mode and testing mode. Figure 6 is a sample timing diagram of testing the network. In training mode, the iris vectors are fed into the network iteratively. The training process is done between 0 ns to 36000 ns. This process is activated by selecting mode "10". The first 100 ns are used to initialize the weights of

the hidden layer neurons and the output layer neurons. This process is done in mode "01". At 100 ns, the training phase for the NN chip is started. Inputs to the input layer neuron paired with a desired output are fed to the network. The desired output is the signature, which is chosen to recognize different iris images. Weights are adjusted to minimize the output error. During the training phase, weights generated by the random weight generator were randomly tested. It has been observed that the value of output error is reduced as more input is fed into the network. The weights are adjusted so that the probability of the next input obtaining its desired output is higher. In the training mode, the input to the network is first normalized before the calculation is performed.

Testing phase is performed between 36100 ns and 36800 ns. This process is done in mode "11". Similar to the training phase, the network must pass through the weights initialization process before testing is performed. The weights of the NN are considered to have reached its local or global minimum at this period. There was no adjustment of weights during testing and thus no output error was calculated. It is observed that the values of the variables stop changing at 36100 ns.

It is also observed that changes in weights are less frequent as the network approaches 36100 ns. The reason behind this is that the network

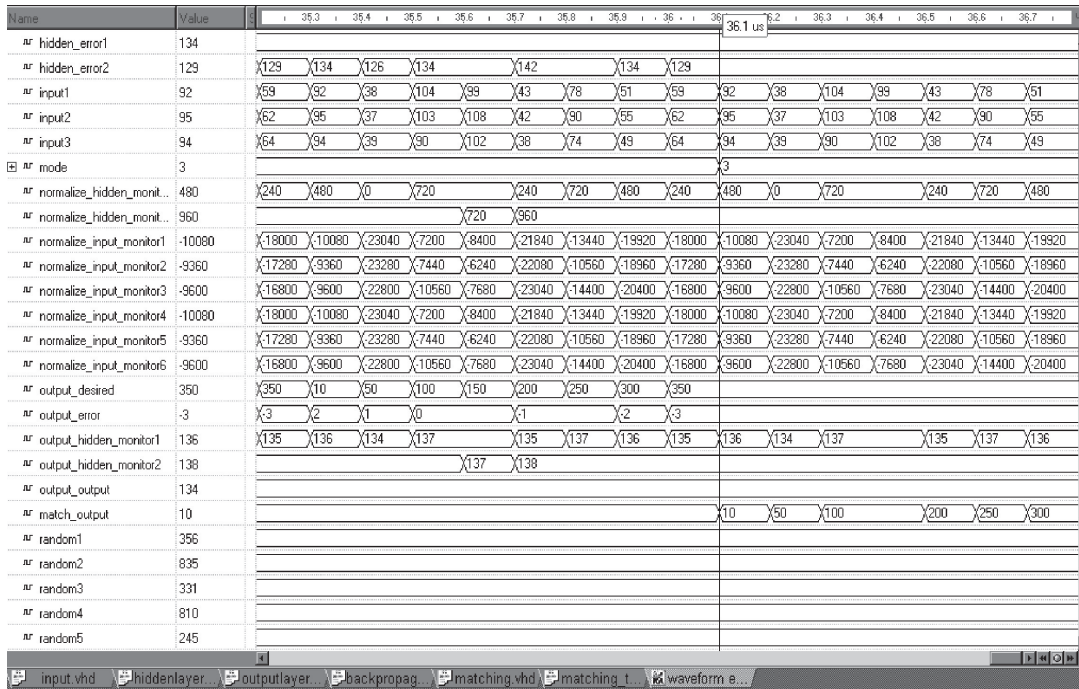


FIGURE 6. Simulation results taken at 36100 ns.

has almost reached its global or local minimum after performing the backpropagation learning algorithm. The simulation is stopped at 36800 ns and the testing phase is completed.

PERFORMANCE COMPARISONS

Hardware Implementation Alternatives

In regard to the designated hardware realization, Table 1 lists the implementation performance and comparison study using a few selected FPGA devices. Each type of FPGA family consists of several devices based on total logic cells and speed.

In optimizing the design, the goal was to choose a balance between frequency/speed (maximum throughput) and area (least gates). These factors need to be taken into account when selecting a suitable device. The project was successfully synthesized using Mercury EP1M120F484C5 device family from Altera Corporation as it utilized the lowest area (smallest number of logic cells consumed) and offered the highest frequency.

From Table 2, it can be seen that the method proposed by Daugman (2001) yielded the highest result. However, the system of Daugman concentrated on ensuring that repeated image captures produced irises on the same location within the image, had the same resolution, and were glare-free under fixed illumination. The work by Ma et al. (2002) employing the use of a bank of spatial filters produced the second highest accuracy at 99.19%. However, the size of the kernel influences the flexibility and precision of the spatial filter. The size of the kernel is limited by the processing capability of the computer system. This method also suffers from a high computation power.

It is also observed that all the methods above are based on software implementation. Often, many practical applications involving recognition systems require large computational power to cope with complexity or real-time constraints. Often such power is not available and is very costly. In addition, software implementation must run on computers or large machines and

TABLE 1. Implementation performances

Family	Device	Area	F _{max} (MHz)
Apex20KC	EP20K200CQ240C7	4158/8320 (49.98%)	84.57
ApexII	EP2A25B724C7	4159/24320 (17.10%)	85.47
Flex10K	EPF10K20RC240-3	4157/1152 (not fitted)	55.56
Mercury	EP1M120F484C5	4157/4800 (86.60 %)	121.87
Apex20ke	EP20K200EFC672-1X	4162/8320 (50.02%)	120.75

Comparison among Related Works

In order to examine and analyze the effectiveness and feasibility of the proposed approach, a comparison with related work in terms of percentage of accuracy and computation time was made and presented in Table 2. The data set that was used in this study is a combination of samples collected from the related works and generated samples.

this poses a constraint on space and makes a classifier system less portable. On the other hand, hardware implementations already provide, for many applications, adequate power and gates in a single chip package (John & Richard 1995). Thus, the method in this study is much more computational and implementation efficient.

TABLE 2. Performance comparisons

Method	Accuracy (%)	Recognition time (ns)
Boles & Boashash (1998)	92.64	11.0
Daugman (2001)	100	4.3
This study	98.5	0.0368
Wildes et al. (1996)	-	401.0
Ma et al. (2002) Vol.I	94.91	7.5
Ma et al. (2002) Vol.II	99.19	13.1

Table 2 also shows an ideal case study to compare speed on software-developed systems and hardware-developed systems by referring to the recognition time produced by the different methods. It is obvious that the method in this study achieved the fastest recognition time of 0.0368 ns. This time refers to the NN recognition time. This validates the claim that hardware implementation provides a higher speed as compared to software simulation.

The method by Daugman (2001) ranked second with a recognition time of 4.3 ns. This is because the method employed can compute the distance between a pair of feature vectors by the XOR operation. If the XOR operation is performed using some optimization schemes in C/C++, the running time for matching can be significantly reduced. That is, Daugman's method is efficient to make one-to-many search in a large-scale database. The method by Wildes et al. (1996)

requires the highest recognition time of 401 ns as the method employed is very complex and computationally demanding. This shows that limitations exist for software implementation in which they will never be fast enough as compared to hardware platforms.

CONCLUSION

An approach of using FPGA for the realization iris recognition system is presented. The performance of the recognition system is encouraging. The results reveal that the proposed approach is computationally simple, and exhibits a good balance of flexibility, accuracy, speed, size and design cycle time. These important characteristics are needed to design a good recognition system. Comparisons and results presented validate the successful implementation of an iris recognition system.

REFERENCES

- Bertillon, A. 1885. La couleur de l'iris. *Rev. Sci.* 36(3): 65-73.
- Boles, W. & Boashash, B. 1998. A Human Identification Technique Using Images of the Iris and Wavelet Transform. *IEEE Trans. on Signal Processing*. 46(4): 1185-1188.
- Daugman, J. 1993. High Confidence Visual Recognition of Persons by a Test of Statistical Independence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 15(11): 1148-1161.
- Daugman, J. 2001. Statistical Richness of Visual Phase Information: Update on Recognizing Persons by Iris Patterns. *International Journal of Computer Vision*. 45(1): 25-38.
- Daugman, J. 2003. Demodulation by Complex-valued Wavelets for Stochastic Pattern Recognition. *International Journal of Wavelets, Multi-resolution and Information Processing*. 1(1): 1-17.
- Flom, L. & Safir, A. 1987. Iris Recognition System. *U.S. Patent*, No. 4641394.
- Florence Choong, Faisal Yasin, Shahiman Sulaiman & Mamun Reaz 2004. VHDL Modeling of an ANN for Classification of PQ Disturbance. *WSEAS Transactions on Systems*. 3(4): 1355-1360.
- Huang, Shyh-Jier, Tsai-Ming Yang & Jiann-Tseng Huang 2002. FPGA Realization of Wavelet Transform for Detection of Electric Power System Disturbances. *IEEE Transactions on Power Delivery*. 17(2): 388-394.
- Jain, A. K., Bolle R. M. & Pankanti S. 1999. Biometrics: Personal Identification in a *Networked Society*, Norwell: Kluwer.
- John V. O. & Richard C. D. 1995. Field Programmable Gate Arrays. *New York: John Wiley & Sons, Inc.*
- Lim, S., Lee, K., Byeon, O. & Kim, T. 2001. Efficient Iris Recognition through Improvement of Feature Vector and Classifier. *ETRI Journal*. 23(2): 61-70.
- Ma, L., Wang, Y. & Tan, T. 2002. Iris Recognition Based on Multichannel Gabor Filtering. *Proc. of the 5th Asian Conference on Computer Vision*, 1: 279-283.
- Ma, L., Wang, Y. & Tan, T. 2002. Iris Recognition Using Circular Symmetric Filters. *Proc. of the 16th International Conference on Pattern Recognition* 2: 414-417.
- Mohd-Yasin, F., Ibne-Reaz, M.B., Sulaiman, M.S., Choong, F. & Aladeen, A. 2003. Design of Digital Residential Energy Meter Employing VHDL. *Proceedings of the IEEE International Conference on Consumer Electronics*, 1127-1130. Sydney, Australia.
- Wildes, R., Asmuth, J., Hsu, S., Kolczynski, R., Matey, J. & McBride, S. 1996. Automated, Noninvasive Iris Recognition System and Method. *U.S. Patent*, No. 5572596.
- Wildes, R., Asmuth, J., Green, G., Hsu, S., Kolczynski, R., Matey, J. & McBride, S. 1996. A Machine-vision System for Iris Recognition. *Machine Vision and Applications*, 9: 1-8.

